(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
18 January 2001 (18.01.2001)

PCT

(10) International Publication Number
WO 01/04775 A2

(51) International Patent Classification[7]: G06F 17/00

(21) International Application Number: PCT/IL00/00417

(22) International Filing Date: 14 July 2000 (14.07.2000)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
09/353,452 14 July 1999 (14.07.1999) US

(71) Applicant *(for all designated States except US)*: **PARAGON B.V.** [NL/NL]; Paardekooper & Hoffman, Amsteldijk 194, P.O. 7266, NL-107 AG Amsterdam (NL).

(72) Inventors; and

(75) Inventors/Applicants *(for US only)*: **BEN PORAT, Tsvika** [IL/IL]; Derech Yatir Street 107, 85025 Meitar (IL). **EREZ, Luz** [IL/IL]; Messada Street 12, 85025 Meitar (IL). **OFEK, Ziv** [IL/IL]; Beri Street 9, 85025 Meitar (IL).

(74) Agent: **REINHOLD COHN AND PARTNERS**; P.O. Box 4060, 61040 Tel-Aviv (IL).

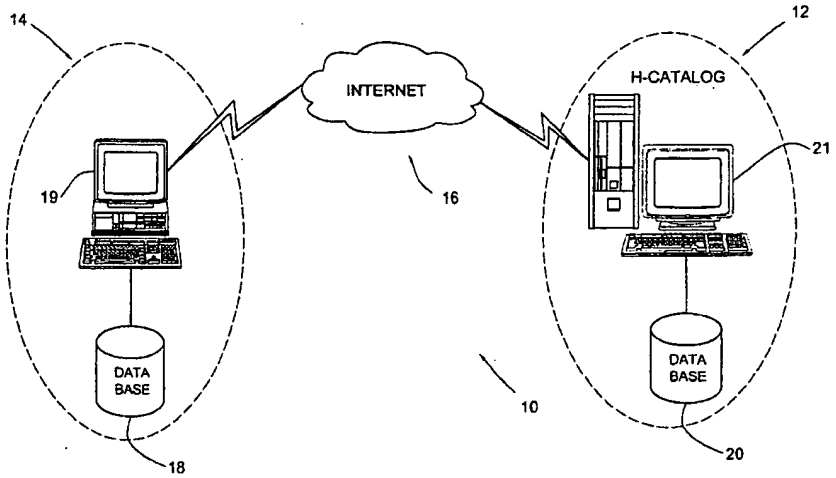
(81) Designated States *(national)*: AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, BZ, CA, CH, CN, CR, CU, CZ, DE, DK, DM, DZ, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, TZ, UA, UG, US, UZ, VN, YU, ZA, ZW.

(84) Designated States *(regional)*: ARIPO patent (GH, GM, KE, LS, MW, MZ, SD, SL, SZ, TZ, UG, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GW, ML, MR, NE, SN, TD, TG).

Published:
— *Without international search report and to be republished upon receipt of that report.*

For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

(54) Title: A METHOD FOR CONSTRUCTING A HOMOGENEOUS ELECTRONIC CATALOG



(57) Abstract: A method for constructing an electronic homogenous catalog database from a plurality of separate suppliers catalog databases, including performing in respect of each separate supplier catalog, the following steps. First, linking the homogeneous database to the supplier database using a communication protocol. Next, mapping selected fields in the supplier catalog database to corresponding fields in the homogenous catalog database. The fields include "field type" fields, "property type" fields and "category type" fields. Next, mapping category values in the supplier catalog database to corresponding category values in the homogenous catalog database. Next, mapping property values in the supplier catalog database to corresponding property values in the homogenous catalog database. Finally, transferring data contained in the fields from said supplier catalog database to the homogenous catalog database.

WO 01/04775 A2

A METHOD FOR CONSTRUCTING A HOMOGENEOUS ELECTRONIC CATALOG

FIELD OF THE INVENTION

This invention relates to the generation and use of electronic catalogs.

BACKGROUND OF THE INVENTION

The amount of textual information that is available in computerized media has increased dramatically in recent years. The wide circulation of the Internet and the provision of a relatively secured transactions (having monetary value) over the Internet has resulted in a flood of electronic catalogs that are offered by suppliers and allow subscribers to visit catalog sites, view products of interest and possibly order them.

Typically, each supplier establishes his/her own catalog by constructing a knowledge base consisting of say a hierarchy of concepts and properties that, to the best of his/her understanding, describe the products that are included in the catalog.

Thus, for example, in a catalog of sport products, a typical concept list includes: *item id* (standing for a key field that uniquely identifies each item) *item name (e.g.* Air Jordan shoe)*; item info (e.g.* sport shoe for running). The concepts include also categories such as sport shoe which include properties, e.g. size, color etc.

A user who seeks to locate a desired product or products and to compare proposals offered by two or more suppliers, can enter the supplier's sites and attempt to locate the product(s) of interest. However, in a typical scenario, the user has no knowledge on the identity of the relevant suppliers (or at least not on

all the relevant suppliers), and hence he/she must conduct the search across the Internet in accordance with the product name and/or possible category or property identifying the product.

To this end, the user invokes a query using a search engine utility (e.g. the known *Yahoo)* and provides query parameters that identify the desired product, category and/or property of interest. In the case that the search engine finds a site or sites that include data that match the sought parameters, the data (e.g. relevant *html* pages) are retrieved and downloaded to the user. The so retrieved data are often sorted by some relevance ranking, which is intended to approximate the degree of relevance of the resulting data to the query.

As is well known to those who try to target specific data in a large knowledge source such as the Internet, the prospects of missing data which reside in the knowledge source and nevertheless are not revealed by the search engine running the query is relatively high, which is obviously undesired. This stems, *inter alia*, from the inherent characteristics of natural language, which enables to define a given concept (e.g. a product), in many different manners.

Thus, when different suppliers make the definitions of products in catalogs separately and independently, a variety of inconsistent definitions are brought about, which, naturally, hamper on the successful targeting of the sought data.

There are known in the art numerous attempts to alleviate the problem, by utilizing sophisticated and very complicated artificial intelligent (AI) based techniques that aim at rendering the numerous dispersed knowledge bases into a harmonized knowledge base structure.

An exemplary (yet not exclusive) AI-based concept harmonization technique includes automatic learning of the "logic" which governs the classification of category. Methods belonging to this approach utilize a set of training data, for which the correct categories are known in advance (usually as the result of manual classification of these categories). A learning method may then include a learning phase, in which some model of the category is constructed. For example, such a model may include terms that are highly

associated with the category, and possibly some weights that quantify the degree of correlation between each term and the category. Alternatively, a learning method may be memory based, in which case the learning method simply stores the training data in some useful format. Then, when a new item (say product) is given for classification, the method classifies it automatically by consulting or applying the category model (or by simply comparing the new data item to the training data, in case of a memory based approach).

However, due to the inherent extremely complex structure of the natural language, these solutions are only partially successful.

There is accordingly a need in the art to provide for a technique which enables to construct a homogenous knowledge base whilst obviating the need to apply complex AI-based techniques.

There is a further need in the art to provide for a technique that enables suppliers to map their respective knowledge base definitions to the specified homogenous knowledge based representation, in a convenient manner utilizing substantially a semi-automatic conversion technique.

GLOSSARY OF TERMS

There follows a glossary of terms some being conventional and others have been coined:

Relational Model (or Database): The relational model, introduced by Codd, is a landmark in the history of database development. In relational databases, an abstract concept has been introduced, according to which the data is represented by tables (referred to as entity or relationship "relations") in which the columns represent the fields and rows represent the records.

The association between tables is only conceptual. It is not part of the database definition. Two tables can be implicitly associated by the fact that they have one or more fields whose values are taken from the same set of values (called "domain").

Other concepts introduced by the relational model are high level operators that operate on tables (i.e. both their parameters and results are tables) and comprehensive data languages (now called 4th generation languages), in which one specifies **what** the required results are, rather than **how** these results are to be produced. Such non-procedural languages (SQL – Structured Query Language) have become an industry standard. Furthermore, the relational model suggests a very high level of data independence. There should not be any effect on the programs written in these languages due to changes in the matter data which are organized, stored, indexed and ordered. The relational model has become a *de-facto* standard for data analysts.

Field: A column in a table of a relational database which represents an attribute of a data record (standing for a product) in the table, for example *color, size, price* in a table that represents *clothing*.. A product is represented by some or all of its fields.

Category: Hierarchical structure of concepts represented as category values, to which products or group of products are classified. In the present invention, products are, typically (although not necessarily) classified to category values taken from the leaf nodes of the hierarchy;

Property: A specific field type which signifies a characteristic of given product or products and which is normally not common to all the products in the catalog. As will be shown below, a property is assigned with property values.

SUMMARY OF THE INVENTION

The terms knowledge base and database are used interchangeably. Whilst for convenience of explanation the invention is described with reference to relational database, the invention is by no means bound to this particular example.

In accordance with the invention, there is provided a technique to construct a homogeneous knowledge base, (constituting a homogeneous catalog) from a

plurality of dispersed knowledge bases, each of which constitutes a separate catalog. In accordance with the invention, for each one of the separate catalogs, all or some of the fields are mapped to corresponding fields in the homogeneous catalog (constituting homogenous catalog field structure). Having constructed the catalog field structure, (which, as will be explained in greater detail below, includes, preferably, fields of "field type", "category type" and "property type"), the category values and property values of the separate catalogs are mapped to the homogeneous catalog. There follows a 'catalog import' step, in which the contents of the supplier database is mapped to the homogenous catalog.

Thus, there is provided in accordance with the invention, a method for constructing an electronic homogenous catalog database from a plurality of separate suppliers catalog databases, comprising performing in respect of each separate supplier catalog, the following steps, that include:

(a) linking the homogeneous database to the supplier database using a communication protocol;

(b) mapping selected fields in the supplier catalog database to corresponding fields in the homogenous catalog database; said fields include "field type" fields, "property type" fields and "category type" fields;

(c) mapping category values in said supplier catalog database to corresponding category values in the homogenous catalog database;

(d) mapping property values in said supplier catalog database to corresponding property values in the homogenous catalog database; and transferring data contained in said fields from said supplier catalog database to said homogenous catalog database.

In accordance with a preferred embodiment, the category values of the source supplier database are mapped to respective category values in the homogenous catalog using, preferably a "group by" function.

In a similar manner, additional separate catalogs are mapped to the same homogeneous catalog. The catalog stores in a unified and homogenous manner, data originated from said separate catalogs.

Preferably, although not necessarily, the knowledge base is arranged in accordance with the relational model database.

In accordance with a preferred embodiment of the invention, there are provided separate catalogs and at least one remote homogenous catalog inter-linked by means of communication network. A typical, yet not exclusive example of a communication network being the Internet.

Having constructed the homogenous catalog in the manner specified, the catalog may be subject to queries, utilizing e.g. conventional query languages such as SQL.

Unlike prior art, where due to inconsistent nomenclature utilized by each separate catalog the queries were subjected to incomplete answers, in accordance with the homogeneous catalog of the invention, the query that is applied to the homogeneous catalog uses terms which are identical (or substantially identical) to those that constitute the homogeneous catalog, and therefore the prospects of missing data due to inconsistent definitions are substantially reduced or even eliminated. The specified terms may be chosen to be field names – (of field, category and/or property type), category values, and /or property values). Other terms may also be used, all as required and appropriate.

BRIEF DESCRIPTION OF THE DRAWINGS

In order to understand the invention and to see how it may be carried out in practice, a preferred embodiment will now be described, by way of non-limiting example only, with reference to the accompanying drawings, in which:

Fig. 1 is a generalized system architecture in accordance with the invention;

Fig. 2 is a flow chart illustrating a generalized sequence of operation, in accordance with the invention;

Fig. 3 is a generalized flow chart illustrating a field mapping sequence in accordance with one embodiment of the invention;

Fig. 4A-B illustrate exemplary user interface screens for realizing the field mapping sequence of Fig. 3;

Fig. 5 is a generalized flow chart illustrating category values mapping sequence in accordance with one embodiment of the invention;

Figs. 6A-B illustrate exemplary user interface screens for realizing the category values mapping sequence of Fig. 5;

Fig. 7 illustrates a typical hierarchy of categories;

Fig. 8; is a generalized flow chart illustrating a property mapping sequence in accordance with one embodiment of the invention; and

Figs. 9A-C illustrates the resulting homogeneous catalog after mapping the fields, category values and property values, represented in an efficient manner, in accordance with one embodiment of the invention.

DETAILED DESCRIPTION OF A PREFERRED EMBODIMENT

The preferred embodiment is illustrated with reference to remote homogenous catalog database and supplier catalog databases, which are linked by the Internet. The invention is by no means bound by this specific example.

Turning now to Fig. 1, there is shown a generalized system architecture in accordance with the invention. The system **(10)** includes a homogenous catalog site **(12)** and a supplier catalog site **(14)**, inter-linked by means of Internet network **(16)**. In accordance with the system of Fig. 1, the separate catalog **(18)** (coupled to conventional desktop **19**) at the supplier site **(14)** is mapped, in a convenient semi-automatic procedure, into homogenous catalog **(20)** (coupled to desktop **21**) at the remote site **(12)**, using a mapping protocol over the internet **(16)**. Whilst in Fig. 1, each catalog is held in one physical storage medium, the invention is not bound to any particular physical and/or logical representation of the catalog sites.

Turning now to Fig. 2, there is shown a flow chart illustrating a generalized sequence of operation, in accordance with the invention. Thus, at the onset, the

supplier **(14)** logs into the homogenous catalog (standing for the server) site **(12)** and after undergoing known *per se* admittance control steps **(30** to **32)**, the server catalog "accesses" the supplier's site (standing for the client), and by means of e.g. known *per se* ODBC driver **(33)** links to the client's database. For convenience of explanation, it is assumed that the catalog database at the client's site is arranged in accordance with the relational model. Those versed in the art will readily appreciate that the invention is by no means bound by any particular high-level or low-level model for representing data. Thus, by way of example, in accordance with one embodiment, a flat model (where all the data is held in one table) is utilized.

Having linked to the client's catalog, the fields **(34)** are mapped (including field type, category type and property type). Having mapped the fields (to thereby constitute a field structure), there follows category values mapping **(35)** followed by property values mapping **(36).**

Having mapped category values and property values, the contents of the supplier catalog is mapped to the homogenous catalog (referred to as catalog import step **(37),** and thereafter integrity checking steps **(38)** and **(39)** are performed, in which errors are rejected **(38)** and an optional manual modification step is provided **(39)** (e.g. for inputting missing data, such as contents of fields, say the value of *color* field for the product *jeans trouser*). The process terminates by providing a status summary report **(39')** (e.g. success or error, and in the latter case also indicating the error type).

Turning now to Fig. 3, there is shown a generalized flow chart illustrating a field mapping sequence in accordance with one embodiment of the invention. Thus, after the ODBC connection **(41),** the supplier identifies the tables which are subject to mapping (and which constitute the supplier's catalog). Considering that the data format and structure of the supplier's catalog are *a priori* known to the communication application, the fields of the catalog table or tables and also the contents thereof can be easily identified, all as known *per se.*

Thus, the fields of the client's catalog are mapped into fields at the server's catalog **(41)**. The fields mapping is then tested **(42** and **43)** and thereafter the catalog and properties **(44** and **45)** are mapped (see below).

For a better understanding of the field mapping sequence, attention is directed to **Figs. 4A-B** that illustrate an exemplary user interface screens for realizing the field mapping sequence of Fig. 3. At the onset, the communication protocol program that links the client and server sites identifies the appropriate relational tables in the server catalog and client catalog that are subject to mapping (known *per se* and not shown). After duly identifying the respective tables, the fields thereof are identified and the list of field names of the table in the homogenous catalog is presented in the left column under the title (*h-catalog field* – where h-catalog stands for homogenous catalog). In Fig. 4A, the catalog is presented as *e-FES*. The supplier now maps manually the field names in his catalog (which, although not shown in Fig. 4A, are normally presented in the right column under the title *Data Source Field*) to the corresponding homogenous catalog fields. The mapped result is shown in Fig. 4B. Thus, the homogenous catalog field *ItemName* is mapped to the corresponding client catalog field name *ProdName*. Likewise, the homogenous catalog field *SizeName* is mapped to the corresponding client catalog field name *CategorName,* and the homogenous catalog field *SmallPicture* is mapped to the corresponding client catalog field name *SmallPic*. As shown, the client is not obliged to map all the existing fields of the catalog into corresponding fields in his local catalog. Put differently, only those fields that are of interest are mapped.

It should be noted that in the process of field mapping, fields of all types are mapped, normally, "fields", "category" and "property" (and if desired possibly also others, all as required and appropriate, depending upon the particular application).

Consider, for example, the following list of fields:

H-Catalog	*Data Source Field (Supplier)*
(1) product name	prod. Name
(2) catalog number	cat. num.
(3) price	Price
(4) category	Category
(5) size	Fit
(6) color	Color

TABLE 1

wherein *H-Catalog* stands for the homogenous catalog side and *Data Source Field* stands for the supplier's fields as appearing in his/her local database. Focusing now on the h-catalog side, the fields 1,2 and 3 stand for "fields", since they are common attributes to all products. Put differently, every product must have a name (field no. 1), a catalog number (field number 2) and a price (field number 3). Field number 4 stands for "category" (as will be explained in greater detail below).

Fields 5 and 6 stand for "property". As specified above, property is, as a rule, an attribute of one or more products, but not of all of them. Thus, *size* and *color* are attributes of some products such as shoes and shirts, but not of others, such as tyres (for cars). The latter may have other properties such as (tyre) *width* and (tyre) *diameter*.

Accordingly, had the catalog product list encompassed not only shoes and shirts, but also tyres, the list of fields would be as follows:

(1) product name

(2) catalog number

(3) price

(4) category

(5) size

(6) color

(7) width

(8) diameter

wherein field nos. 7 and 8 stand for the newly added properties.

Having mapped the fields (including categories and properties), the categories values are mapped in a similar manner as illustrated for example in Fig. 5. After the category mapping step **(51)**, the appropriate data integrity checking is performed (**52** and **53**).

Category value mapping is illustrated in Figs. 6A and 6B. In Fig. 6A, the category list as extracted from the table of the client catalog is displayed (in the left column under the title *Supplier Category*), and is mapped manually to corresponding category value (in the right column under the title *h- category* – designated also as *e-FES category*) in the homogenous catalog at the server site. The resulting mapping is shown in Fig. 6B. For example, the *h-category* category value *Teamclothing trousers/men* is mapped to the supplier (client) category value *Trousers long Unisex/men.*

For a better understanding of the category value mapping, attention is drawn to Fig. 7, illustrating the hierarchy (tree) of categories **(70)**, of the supplier end.

The hierarchy tree is a non-limiting form of representing categories. In the hierarchy tree, the most generalized definition of categories resides (at the top of the hierarchy –referred to also as *root* node) and more specific definitions reside in lower levels of the tree.

Thus, for example, (see Fig. 7) root **(71)** represents the general category definition *clothing.* Nodes **(72)** and **(73),** lower in the hierarchy, represent more specific category definition (*youth clothing* and *kids' clothing*). The category values residing at the lowest level of the hierarchy (leave nodes) represent the most

specific category definition. Thus, **(74)** represents *sport pants for youth*, and category value **(75)** represents *elegant pants for youth*. Note that a category value may be viewed as concatenation of the nodes from root to leaf. Thus *clothing -> youth -> pants -> sport* corresponds to category value **(74).**

In order to map the category values, it is first required to "flatten" the hierarchical representation of categories in order to obtain a list of category values that are subject to mapping (from the *supplier* catalog to the *h-catalog* database). The flattening results in extracting only the category values of interest, and in the specific example of Fig. 7, this means category values (**74** to **78**). Whilst in the specific example of Fig. 7 only leaf nodes were extracted (for category value mapping purposes), this is not necessarily always the case. Thus, by an alternative embodiment, higher levels in the hierarchy of categories may also or alternatively be used. The flattening procedure is implemented manually, or in a semi-automatic manner. As explained above, the mapping of category values is implemented basically in the same manner as mapping the fields.

Reverting now to the latter example, the mapping results of some of the category values is illustrated in table 2 below:

H-CATALOG	*SUPPLIER*
Sport pants for youth	Sport pants for youth
Jeans trousers for youth	Jeans trousers for youth
Swimsuits and light Summer clothing for youth	Swimsuits for youth

TABLE 2

In those applications where the categories form an integral part of the product table (of the catalog at the supplier's end), a preliminary "group by" function may be utilized in order to improve the efficiency of the category values mapping. Thus, consider for example a catalog at the supplier end that holds X items all classified to the *Sport pants for youth* category value and additional

Y items all classified to the *Jeans trousers for youth* category value. Obviously, if a single table holds both the item records and the category values to which the items belong, it is expected that under the *category* field, the table includes *X* repetitions of the *Sport pants for youth* category value and *Y* repetitions of the *Jeans trousers for youth* category value.

Following a naive category mapping sequence may lead to redundant mapping of the same category value again and again (X times for the *Sport pants for youth* category value and Y times for the *Jeans trousers for youth* category value).

Applying a known *per se* "group by" function will cope with the specified in-efficient procedure since it delivers as an output only the different category values (and in the latter example only two category values, i.e. *Sport pants for youth* and *Jeans trousers for youth*), and thereby avoid undesired repetitions.

Having mapped the category values (**81** in Fig. 8), there follows a property value mapping step **(82),** followed by conventional property value checking (**83** and **84**).

Reverting now to the previous example, the values of the property "color" are mapped as follows:

H-CATALOG	*SUPPLIER*
green	green
blue	blue
yellow	yellow/blue

TABLE 3

and, the values of property "size" are mapped as follows:

H-CATALOG	*SUPPLIER*
40	One size
38	38
32	32

TABLE 4

Similar to the category mapping, also the property mapping may involve a preliminary "group by" function in order to extract unique property values and avoid repetitions.

Having mapped the fields, the category values and the property values, the contents of the catalog at the supplier end may now be imported to the *h-catalog* database (step **37** in Fig. 2) so as to construct the *h-catalog* catalog database. The data import is realized using known *per se* data transfer techniques. Of course, the data in the server catalog are organized under the field names of the homogeneous catalog.

The original catalog (at the *supplier* site, Table 6), and the mapped catalog (at the *h-catalog* site, Table 5) are, accordingly, as follows:

COLOR	SIZE	CATEGORY	PRICE	CATALOG NUMBER	PRODUCT NAME
Green	40	Sport pants for youth	23	123	Bermuda shorts
Blue	38	Jeans trousers for youth	390	456	Jeans 501
Yellow	32	Swimsuits and light summer clothing for youth	80	789	Swimsuit

TABLE 5

COLOR	SIZE	CATEGORY	PRICE	CAT. NUM.	PROD. NAME
Green	One size	Sport pants for youth	23	123	Bermuda shorts
Blue	38	Jeans trousers for youth	390	456	Jeans 501
Yellow/Blue	32	Swimsuits for youth	80	789	Swimsuit

TABLE 6

For convenience of explanation only three products are shown in tables 5 and 6 above.

Those versed in the art will readily appreciate that the catalog representation as a single table is made for illustrative purposes only and, accordingly, any known *per se* technique for efficiently storing the data, is applicable.

Figs. 9A-C illustrate one out of many possible variants for representing data in the h-catalog site in accordance with one embodiment of the invention.

Thus, for example the entity "fields" **(91),** (Fig. 9A) contains the fields *"product name", "catalog name"* and *"price",* as well as their contents, (i.e. *Bermuda shorts, Jeans 501, Jeans, LEE)* and *Swimsuit* – together with their respective catalog numbers and prices). One from among these fields (or possibly other fields) serves as a key field (*product key)*, all as known *per se.* The category field is represented as an integral part of table **(91).** Table **(92)** (Fig. 9B) stands for category table and it includes the key field *Category Id* and *category name*. The contents of the category table is, as shown, the distinct category values, i.e. "*Sport pants for youth*", "*Jeans trousers for youth*", and "*Swimsuits and light summer clothing for youth".*

Table **(93)** Fig. 9C stands for "property" having "color" and "size" properties and their respective values *green, blue* and *yellow* (for color) and *40, 38* and *32* for size.

The representation in accordance with Figs. 9A-C avoids duplicating the "*category name*" data which is relatively large and duplicating only the compacted category id data.

The representation of data in accordance with Fig. 9A-C is, of course, only one out of many known *per se* manners of representing data and by way of alternative non limiting embodiment the known *ERD* model may be used. As is well known, the latter enables efficient 1:N relationship (e.g. a category can be assigned to more than one product) and N:M representation.

The procedure described with reference to a supplier catalog database is not bound to any specific order or scope. Thus, for example, the entire database may be mapped in one time or, if desired, the procedure described above may be applied successively to database portions, e.g. applied to each database table separately.

The procedure described with reference to Figs. 1 to 9 is repeated for each supplier who wishes to subscribe to the homogenous catalog. Thus, the data of all the separate catalogs are represented in a unified manner in the homogenous catalog and, accordingly, querying the homogenous catalog using the common field, category and/or property nomenclature, will bring about consistent results as compared to the alternative of querying the inconsistent separate catalogs of the suppliers. The actual representation of data in the *h-catalog* and *suppler* may be one in any known *per se* manner taking in account depending on e.g. volume and performance considerations.

Alphabetic characters and roman symbols used to designate method steps are used for convenience of explanation only and do not necessarily imply any particular order steps.

The present invention has been described with a certain degree of particularity, but those versed in the art will readily appreciate that various alterations and modifications may be carried out without departing from the scope of the following claims:

CLAIMS:

1. A method for constructing an electronic homogenous catalog database from a plurality of separate suppliers catalog databases, comprising performing in respect of each separate supplier catalog, the following steps, that include:

(a) linking the homogeneous database to the supplier database using a communication protocol;

(b) mapping selected fields in the supplier catalog database to corresponding fields in the homogenous catalog database; said fields include "field type" fields, "property type" fields and "category type" fields;

(c) mapping category values in said supplier catalog database to corresponding category values in the homogenous catalog database;

(d) mapping property values in said supplier catalog database to corresponding property values in the homogenous catalog database; and

(e) transferring data contained in said fields from said supplier catalog database to said homogenous catalog database.

2. The method of Claim 1, wherein the homogenous database and said plurality of supplier databases being each a relational database.

3. The method of Claim 1, wherein said step (c) includes the following preceding step:

grouping category values in the supplier catalog database so as to obtain unique set of category values.

4. The method of Claim 2, wherein said step (c) includes the following preceding step:

grouping category values in the supplier catalog database so as to obtain unique set of category values.

5. The method of Claim 1, wherein said step (d) includes the following preceding step:

grouping property values in the supplier catalog database so as to obtain unique set of property values per property.

6. The method of Claim 2, wherein said step (d) includes the following preceding step:

grouping property values in the supplier catalog database so as to obtain unique set of property values per property.

7. The method according to Claim 1, wherein said steps (b) to (e) steps are applied separately in respect of each database portion.

8. The method according to Claim 7, wherein said database portion being a database table.

9. The method according to Claim 1, wherein the communication protocol that is used for linking the homogeneous database to the supplier database utilized ODBC driver.

10. The method according to Claim 1, wherein said linking step is accomplished from a remote homogeneous database to the supplier database using communication protocol over a communication network.

11. The method according to Claim 10, wherein said communication network being the Internet.

12. A storage medium containing a homogenous catalog database produced in accordance with the method of Claim 1.

13. A query language utility for querying homogenous catalog database produced in accordance with the method Claim 1.

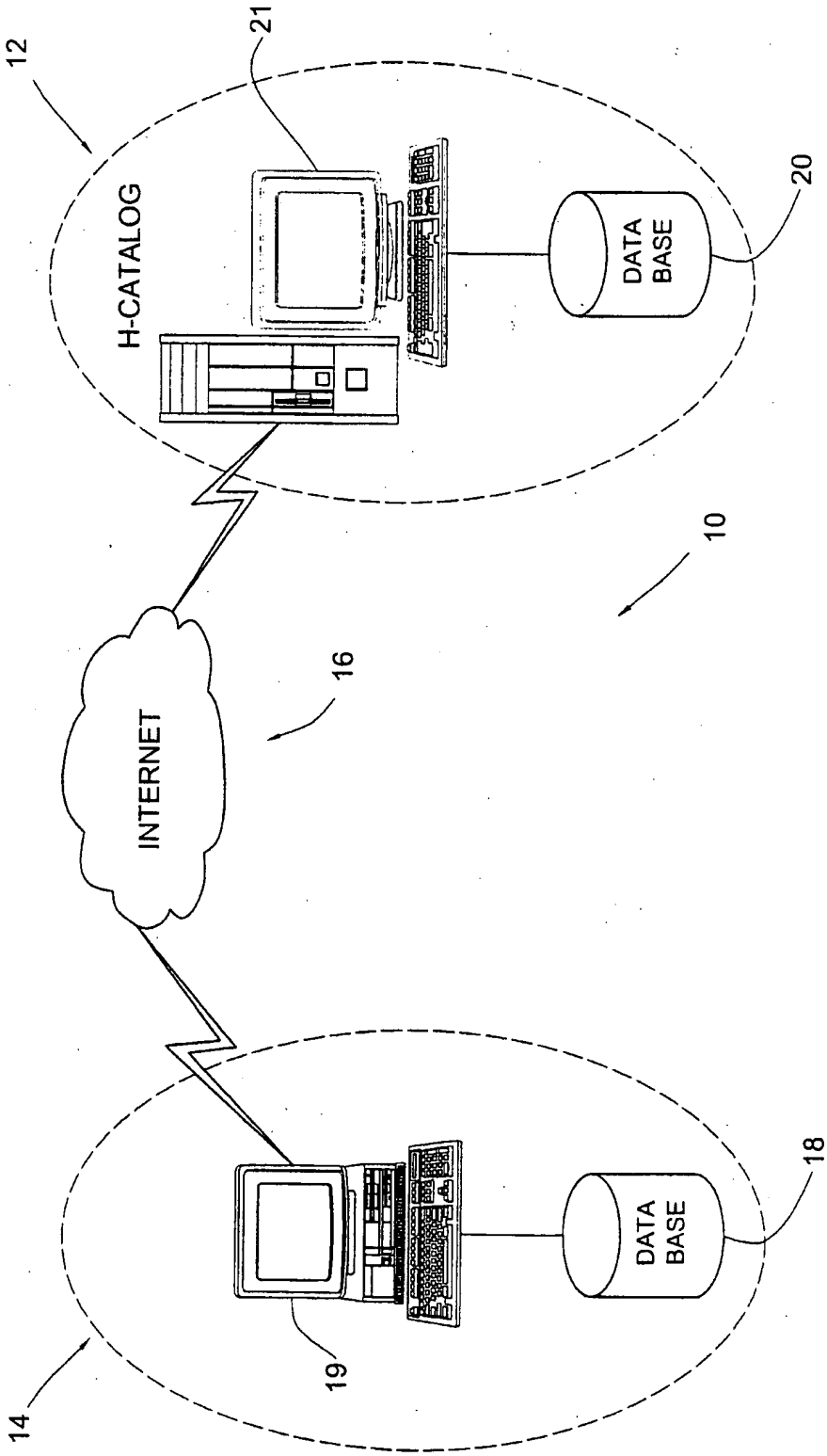


FIG. 1

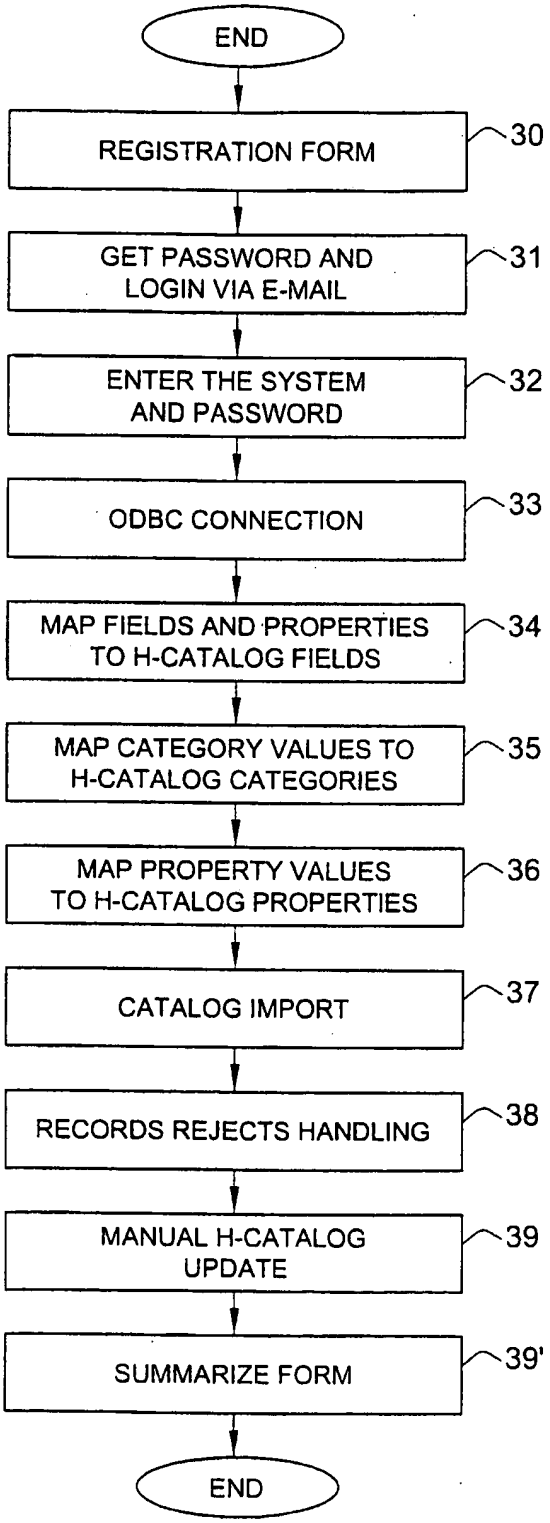


FIG. 2

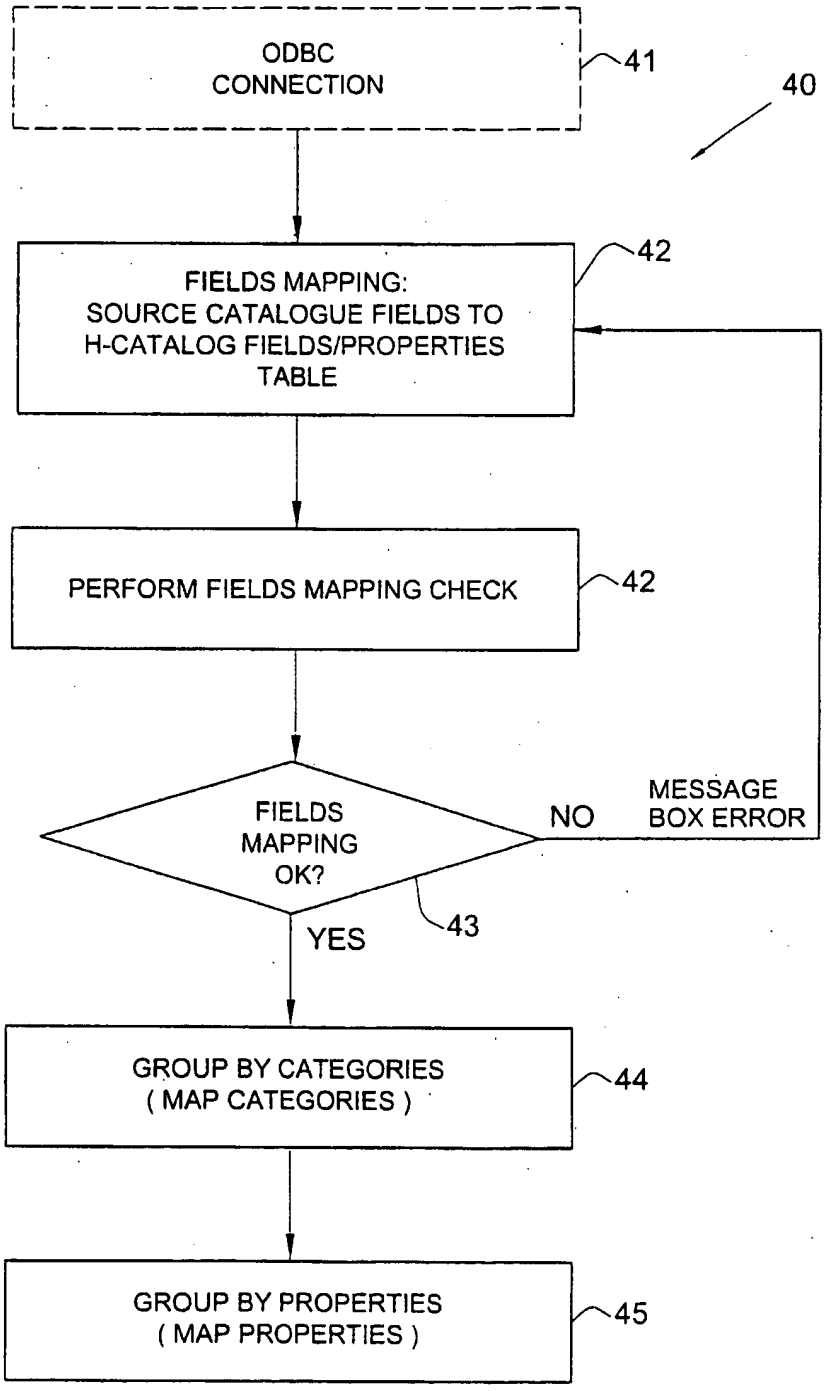


FIG. 3

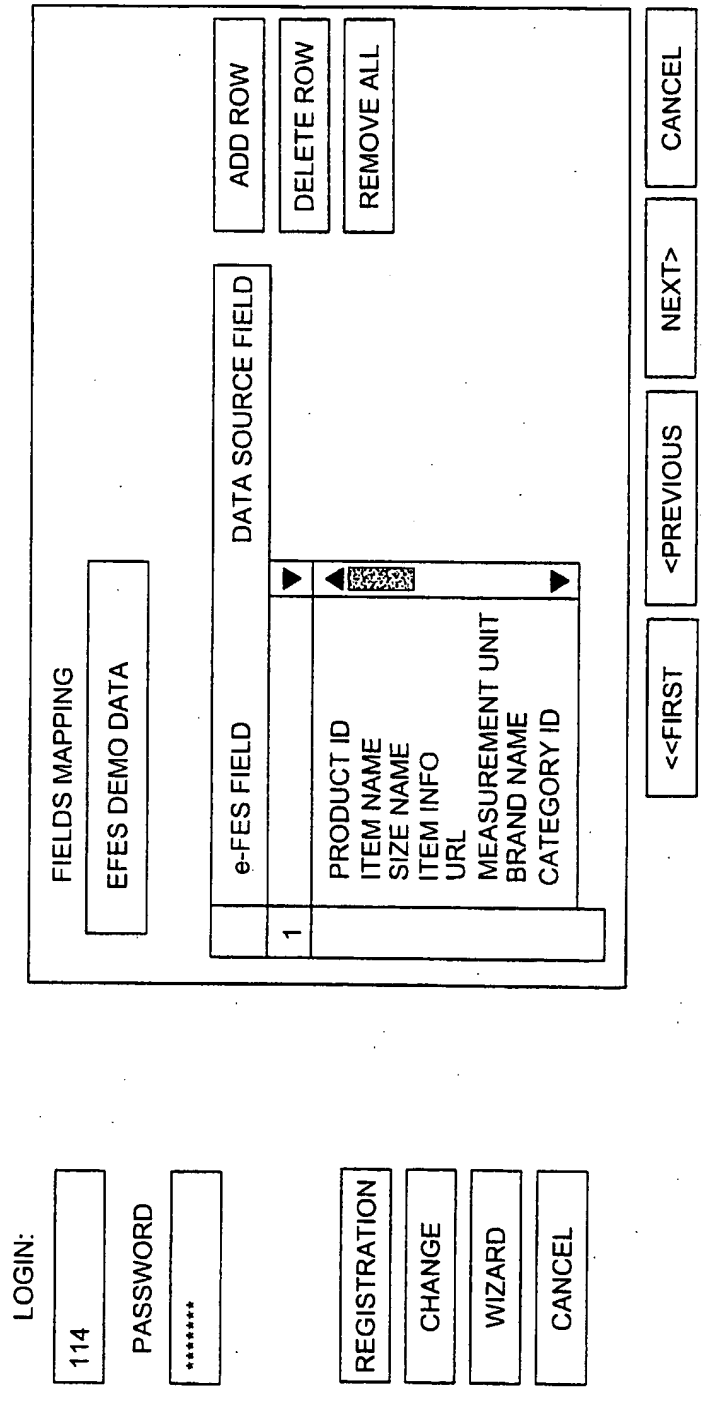


FIG. 4A

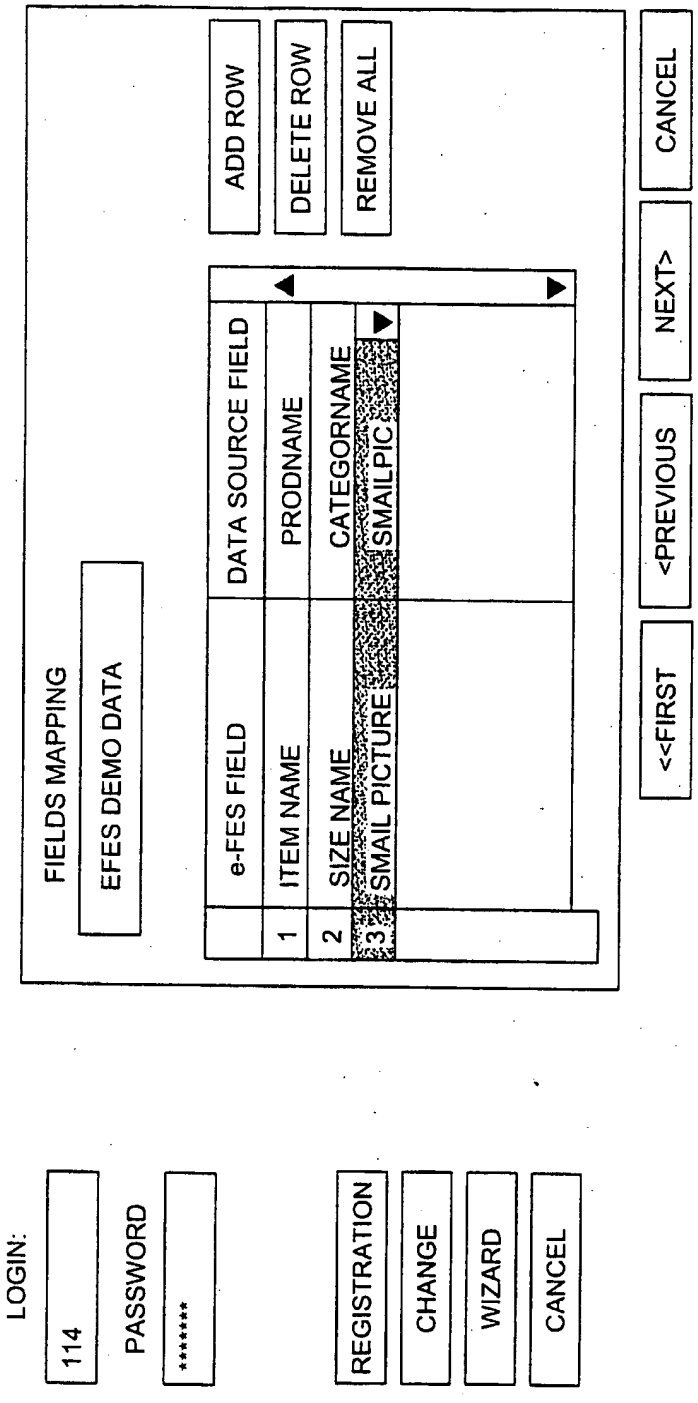

FIELDS MAPPING
EFES DEMO DATA
e-FES FIELD
DATA SOURCE FIELD
1 ITEM NAME PRODNAME
2 SIZE NAME CATEGORNAME
3 SMAIL PICTURE SMAILPIC
ADD ROW
DELETE ROW
REMOVE ALL
<<FIRST
<PREVIOUS
NEXT>
CANCEL
LOGIN:
114
PASSWORD
****
REGISTRATION
CHANGE
WIZARD
CANCEL

FIG. 4B

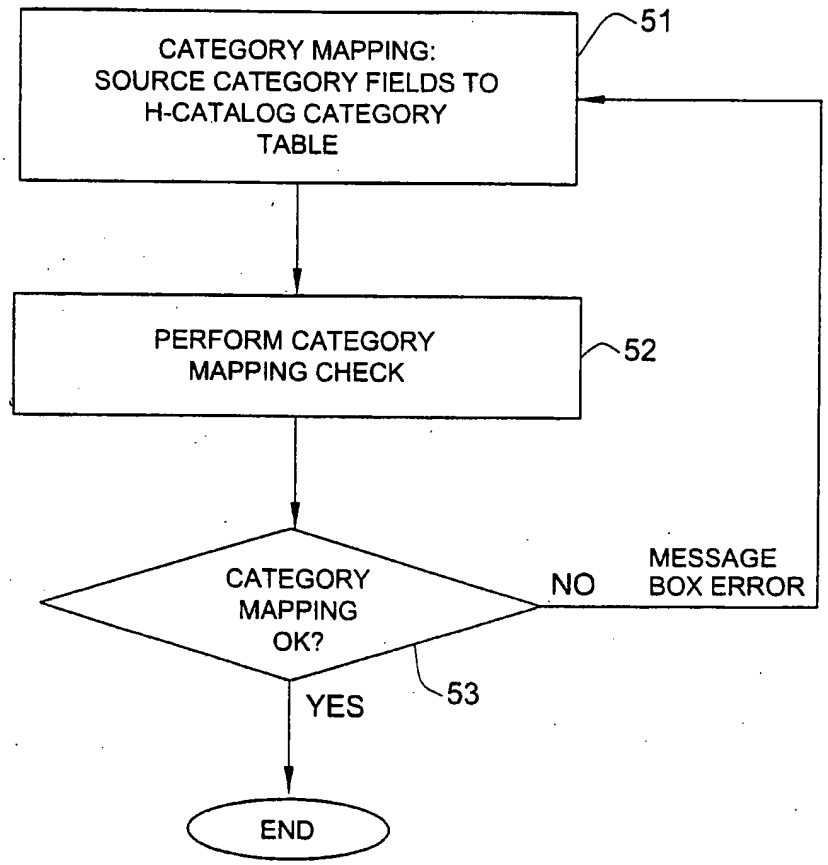

CATEGORY MAPPING: SOURCE CATEGORY FIELDS TO H-CATALOG CATEGORY TABLE
51
PERFORM CATEGORY MAPPING CHECK
52
CATEGORY MAPPING OK?
NO
MESSAGE BOX ERROR
53
YES
END

FIG. 5

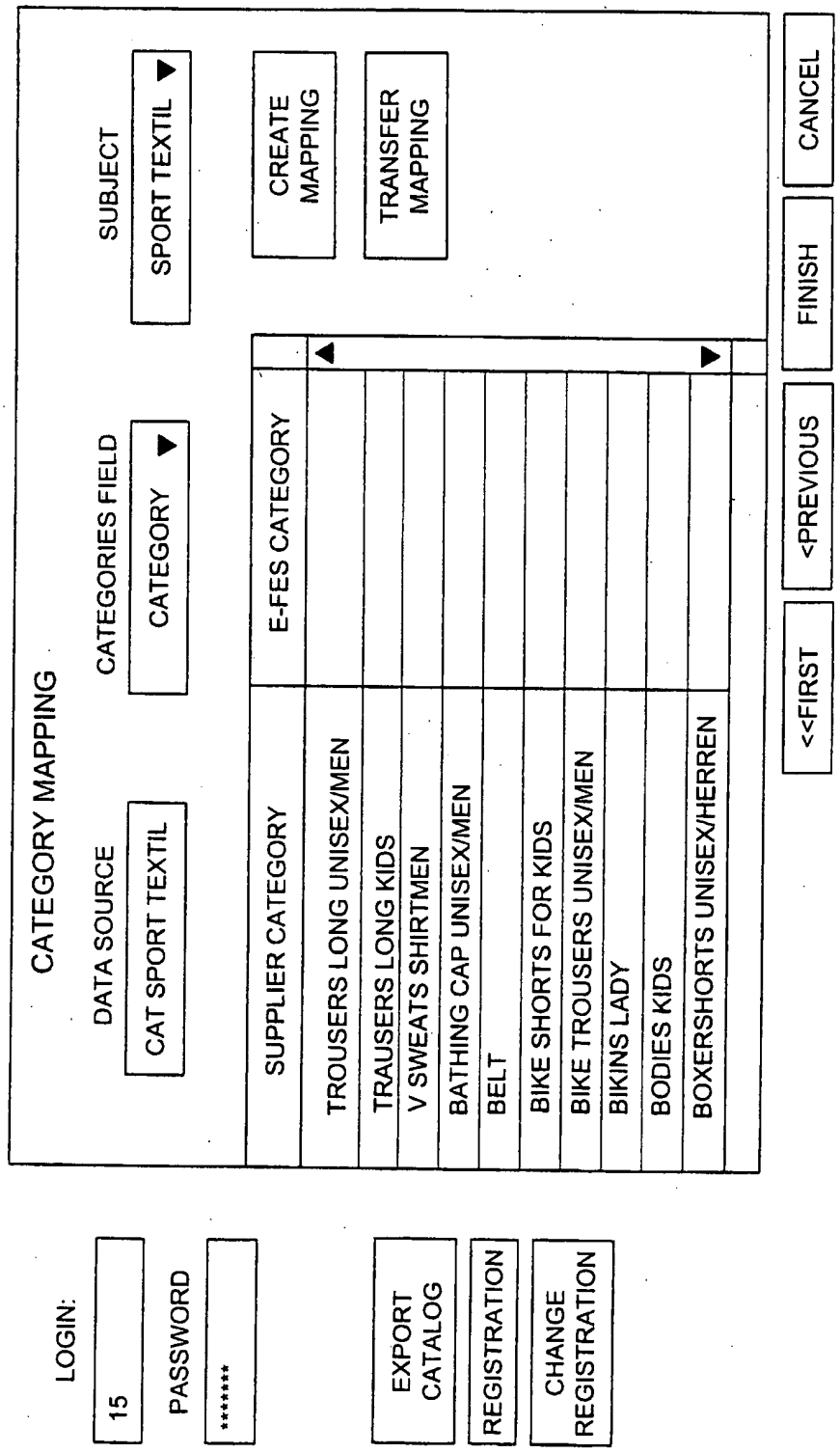


FIG. 6A

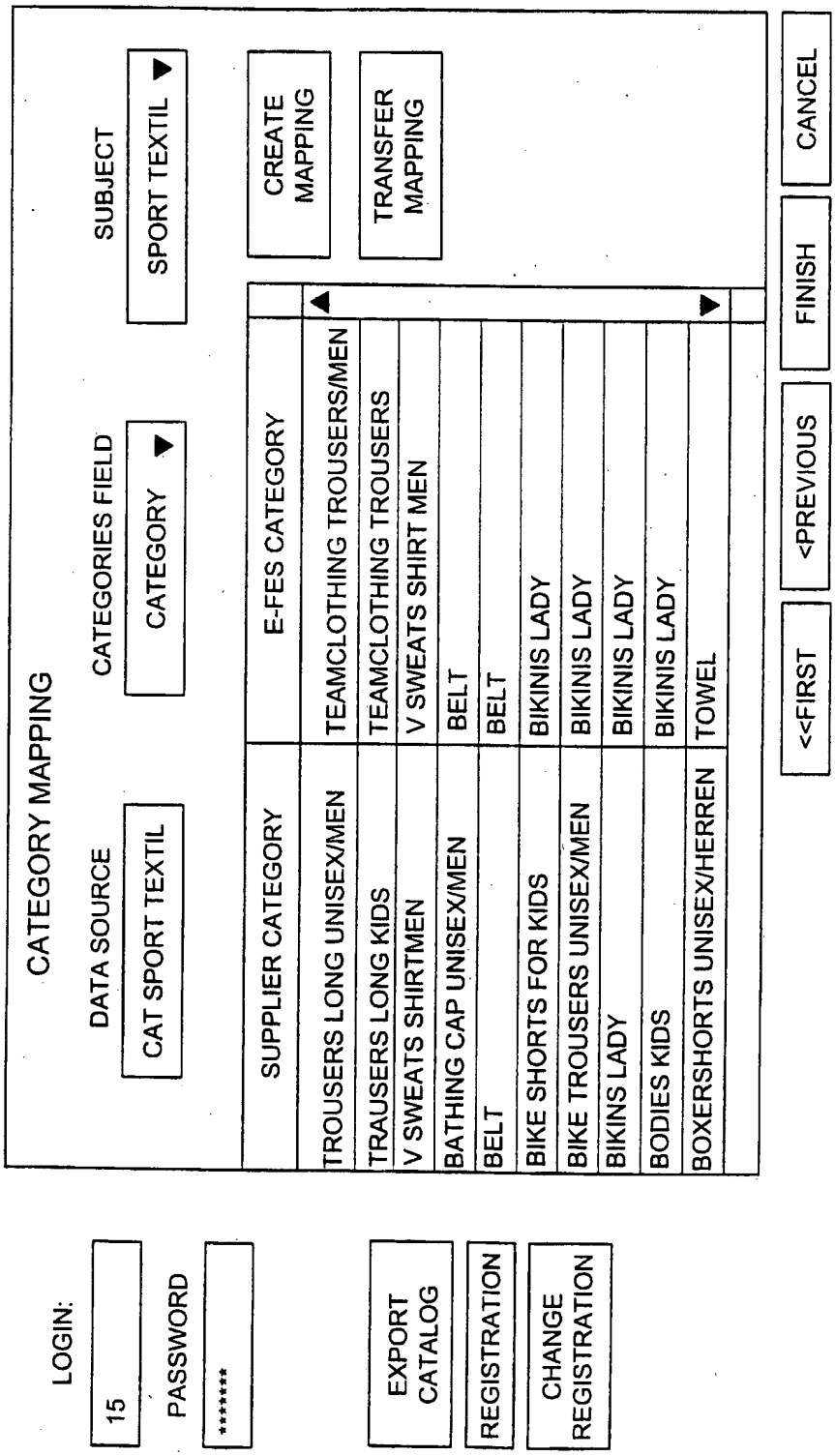


FIG. 6B

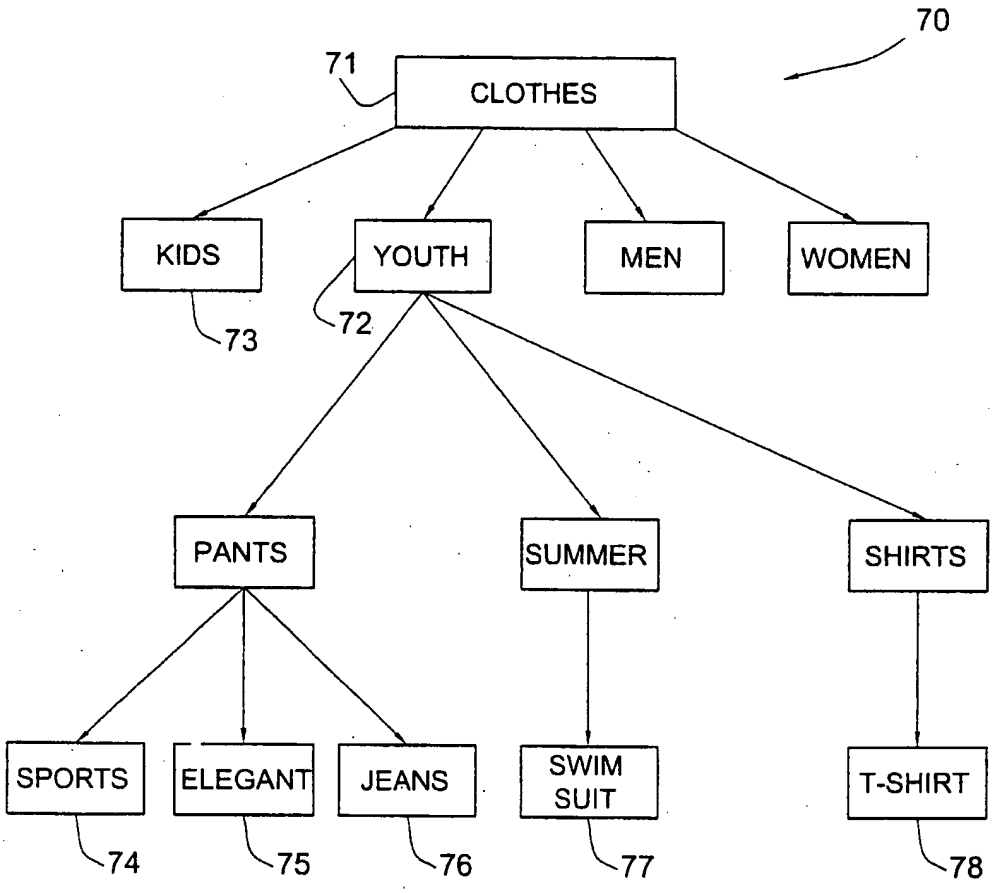


FIG. 7

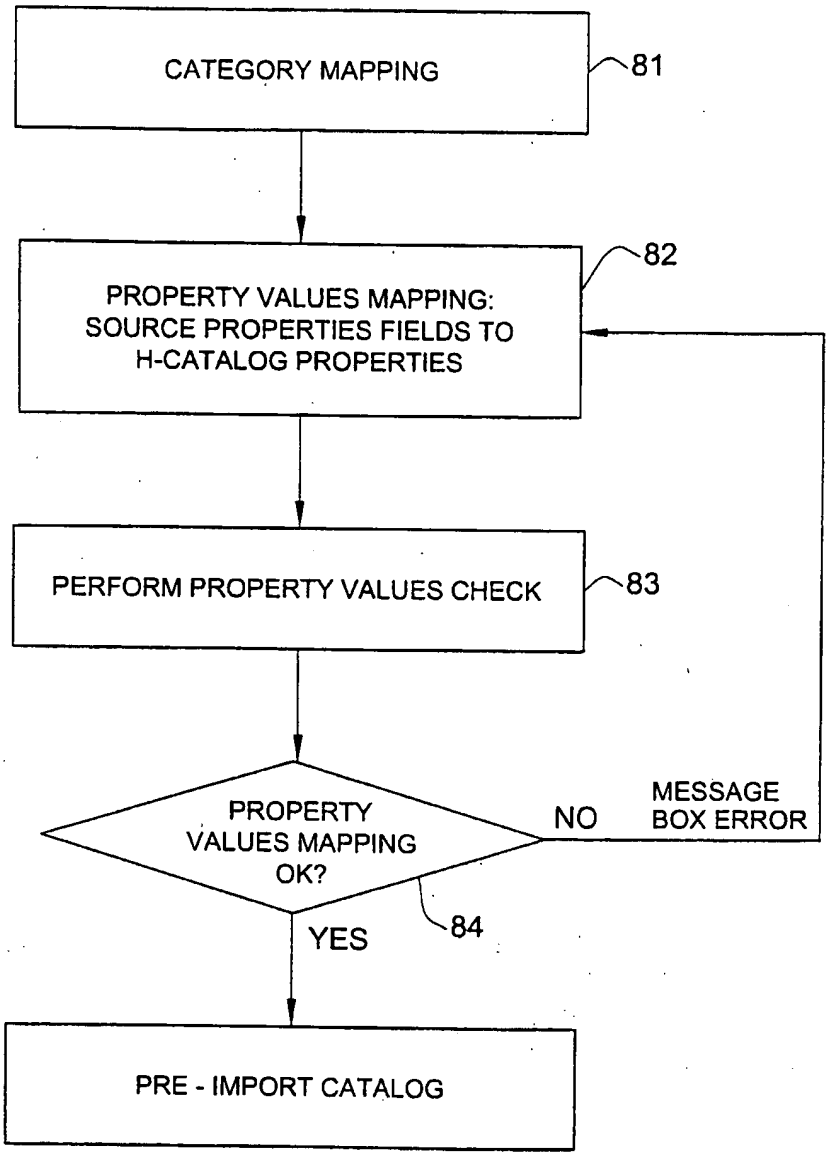

CATEGORY MAPPING
81
PROPERTY VALUES MAPPING: SOURCE PROPERTIES FIELDS TO H-CATALOG PROPERTIES
82
PERFORM PROPERTY VALUES CHECK
83
PROPERTY VALUES MAPPING OK?
NO
MESSAGE BOX ERROR
84
YES
PRE - IMPORT CATALOG

FIG. 8

91

PRODUCT KEY	PRODUCT NAME	CATALOG NUMBER	PRICE	CATEGORY ID
1	BERMUDA SHORTS			1
2	JEANS 501			2
3	JEANS LEE			3
4	SWIM SUIT			4

FIG. 9A

92

CATEGORY ID	CATALOG NAME
1	SPORTS PANTS FOR YOUTH
2	JEANS TROUSERS FOR YOUTH
3	SWIM SUIT AND LIGHT SUMMER CLOTHING FOR YOUTH

FIG. 9B

93

PRODUCT KEY	PROPERTY	PROPERTY VALUE
1	COLOR	BLUE
1	SIZE	38
2	COLOR	BLUE
2	SIZE	38
3	COLOR	BLUE
3	SIZE	38
4	COLOR	GREEN
4	SIZE	38

FIG. 9C